ARMY RESEARCH LABORATORY

An Economical System for Measuring, Monitoring, and Controlling the Rate of Repetitive Events

Thomas Kottke

ARL-MR-180 September 1994

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

NOTICES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1994	Final, Dec 93 - Mar 94

4. TITLE AND SUBTITLE
An Economical System for Measuring, Monitoring and Controlling the Rate of Repetitive Events

5. FUNDING NUMBERS
PR# DC05/H0840

6. AUTHOR(S)
Thomas Kottke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WT-WD
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-OP-AP-L
Aberdeen Proving Ground, MD 21005-5066

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARL-MR-180

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An integrated hardware/software system that can economically measure, monitor, and control the frequencies of repetitive events is presented. A user-adjustable counting interval allows high-resolution frequency measurements to be obtained for repetitive events with frequencies that range from tens of hertz to several kilohertz. The flexibility of this system is further enhanced by the inclusion of microprocessor control that allows both frequency values and trends to be determined and displayed. An assortment of digital inputs and outputs is available to interface this system with external devices such as alarms and frequency feedback controllers. Schematic diagrams, circuit descriptions, printed circuit board masks, and a parts lists of the hardware are included. Sample software that can be used to drive the microprocessor is also presented and discussed.

14. SUBJECT TERMS
frequency, frequency measurement, frequency control, frequency meters

15. NUMBER OF PAGES
47

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Intentionally Left Blank

ACKNOWLEDGMENTS

The author would like to thank Mr. Lawrence W. Burke, Jr. of the Advanced Munitions Concepts Branch (AMCB) of the U. S. Army Research Laboratory (ARL) for providing the means for fabricating printed circuit boards at a time when funding was severely limited. In addition, the author would like to thank Dr. Paul Berning of the Survivability Concepts Branch (SCB), Mr. Wallace H. Clay of AMCB, and Mr. Eric Edwards of LB&B for reviewing and improving this manuscript.

Intentionally Left Blank

TABLE OF CONTENTS

Intentionally Left Blank

LIST OF FIGURES

LIST OF TABLES

Intentionally Left Blank

1. INTRODUCTION

The ability to measure, monitor, and control the frequency of a repetitive event is a common requirement in many laboratory and applied environments. This report presents a detailed description of an economical integrated hardware/software system that has been developed to perform these functions. Originally, this system evolved out of the need to economically monitor and control the rotational frequency of optical chopper wheels. A common method for registering each revolution of such a wheel is to include a single tab on the outer radius that is configured to pass between a light source and a photosensitive detector. As the tab passes between the light source and the sensor, the light beam is blocked and the additional revolution is denoted. For ease of application, the necessary hardware has been included to drive an infrared (IR) light-emitting-diode (LED) and process signals from a PIN diode photosensitive detector. However, the application of this system is not limited to scenarios that involve the physical breaking of a light beam. Indeed, any repetitive activity that indicates each cycle by the change in state of a digital signal can be considered.

A considerable amount of flexibility has been built into this system by the inclusion of microprocessor control. In addition to handling the actual counting operations and updating a numeric display, this microprocessor monitors and controls a number of digital inputs and outputs. Digital inputs are used to count events and alter the microprocessor mode of operation. The processing capability of the microprocessor is then used to calculate and display frequency trends and control external devices via the digital outputs. Thus, the use of microprocessor control allows this system to not only be applied to a wide variety of uses but also to serve multiple roles in each application.

This report includes an extensive presentation of the system's hardware and software. The main text includes schematic diagrams, circuit descriptions, and an explanation of the software that is required to drive the hardware. Appendix A contains

fabrication details such as printed circuit board masks and a parts list. The particular software that was used to monitor chopper wheel rotational frequencies is listed and documented in Appendix B.

2. SYSTEM HARDWARE

2.1 Hardware Overview. The heart of the electronic hardware for this system is the 16C55 microcontroller manufactured by Microchip Technology Incorporated.* This very large scale integrated component is powerful enough to be considered a stand-alone computer on a single integrated circuit (IC) (Microchip Technology Inc. 1992). The features of this microcontroller include 512 words of erasable program read-only memory (EPROM), 24 words of 8-bit random access memory (RAM), 8 special function registers, and clock speeds of up to 20 MHz. Thus, this IC can contain its own instructions (which do not disappear when the power is turned off but can be changed when required), temporarily store and manipulate transient data, count pulses, input data, output data, and do it all in a hurry (and all this at a price that is less than the author's hourly salary!). Anyway, the 16C55 microcontroller coordinates the functions of the other three major hardware subsystems--the processing circuitry for the repetitive input signals, the timing circuitry that determines the amount of time repetitive pulses are to be tallied for, and the numeric display circuitry. Each of these three subsystems will now be considered in detail with emphasis placed on the way the subsystems interface with the microcontroller. Then the microcontroller will be reconsidered in detail, which will naturally lead to a discussion of the software that is used to drive the microcontroller. All discussions of electronic circuitry will refer to Figure 1, which is a complete schematic of the system hardware.

2.2 Input Signal Processing. The input signal processing circuitry includes the provisions that have been made to power a light source and obtain signals from the photosensitive detector that the light source illuminates. Recall that the blocking of this

* The Microchip name is a registered trademark of Microchip Technology Inc., 2355 W. Chandler Blvd., Chandler, AZ 85224-6199, (602) 786-7200.

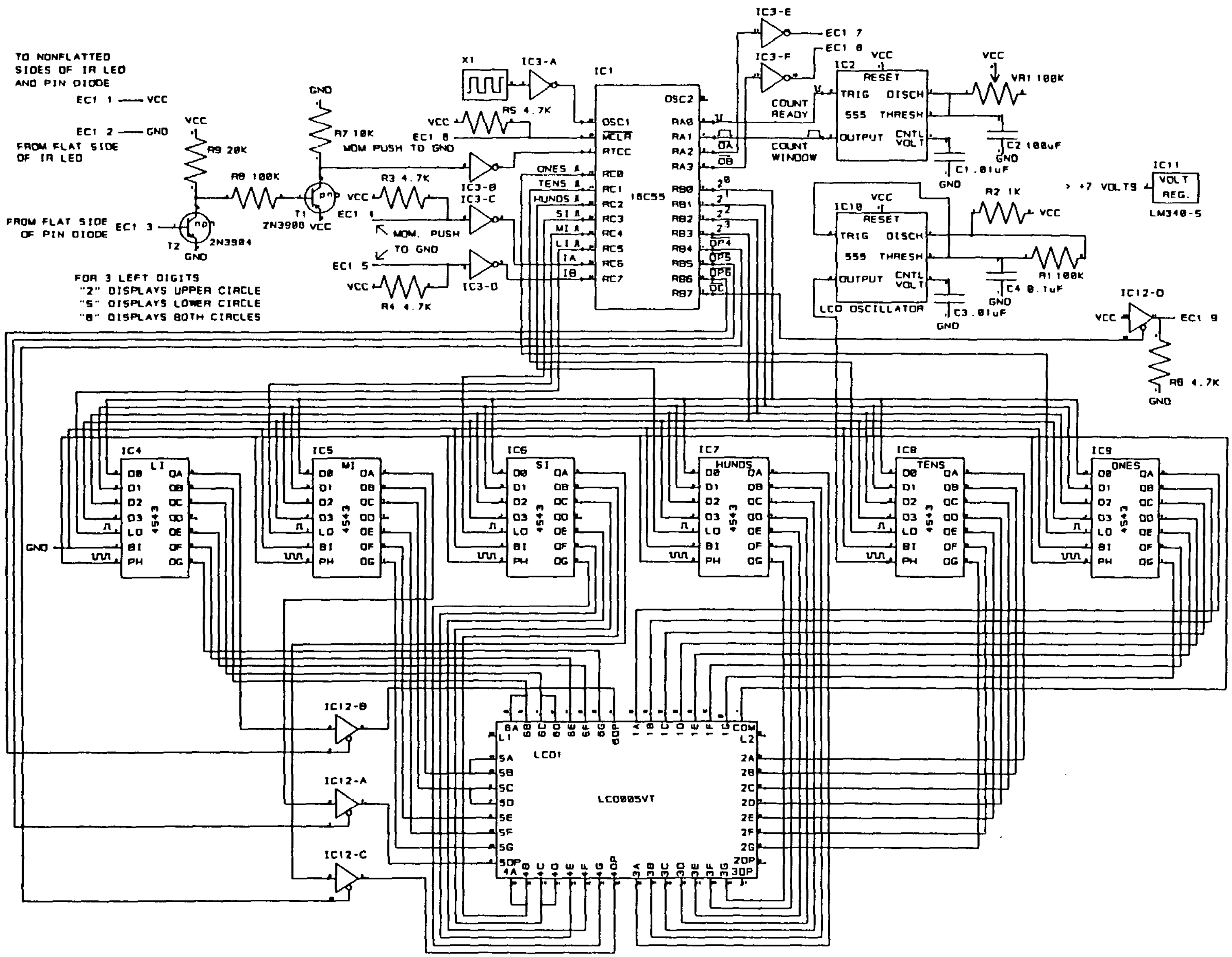


Figure 1. Electronic schematic of the frequency monitoring hardware.

light path can be used to denote repetitive events. The light source that is used in this application is an IR LED which requires little to drive it other than a voltage source in excess of 1.2 V and a series resistor to limit the current through the LED to less than 50 mA. Therefore, the hardware provisions for this light source are minimal. Connector EC1 1 supplies +5 V to the LED which is referenced to the ground potential at EC1 2. It is assumed that the current limiting resistor will be connected to the external LED. Thus, the IR LED light source is wired as shown in the left portion of Figure 2. For this specific application, a 1-kΩ current-limiting resistor was used to limit the LED current to a conservative and power-saving 5 mA.

The circuitry that processes the signal from the PIN diode photosensitive detector is only slightly more involved. PIN diodes get their name from their structural configuration--a high-resistance intrinsic layer of semiconductor material that is sandwiched between a layer of p-type semiconductor and a layer of n-type semiconductor. When a potential is applied across this structure, the high resistance of the intrinsic layer ensures that a large electric field will be associated with this central region. In particular, if this device is reverse biased (a negative potential applied to the p-type material relative to a positive potential to the n-type material), the charge carriers will be swept away from the intrinsic layer, leaving a depletion region that will not allow bias current to flow through this junction. However, if light of sufficient energy is incident on the PIN junction, then electron-hole pairs will be produced and swept from the intrinsic material by the local electric field. Thus, light on the reverse-biased PIN diode produces an electric current. Because these currents are small, they need to be amplified. This is why the signals from the PIN diodes need to be processed.

Transistors are current-magnifying devices and so they are used to amplify the current generated by the PIN diode. The +5-V potential at connector EC1 1 is used to reverse bias the PIN diode by connecting this potential to the diode's cathode. The anode is routed through connector EC1 3 to the base of an npn transistor, which is configured as a common emitter amplifier. The output of this transistor is passed through a second stage of transistor amplification to sharpen up the signal's transition

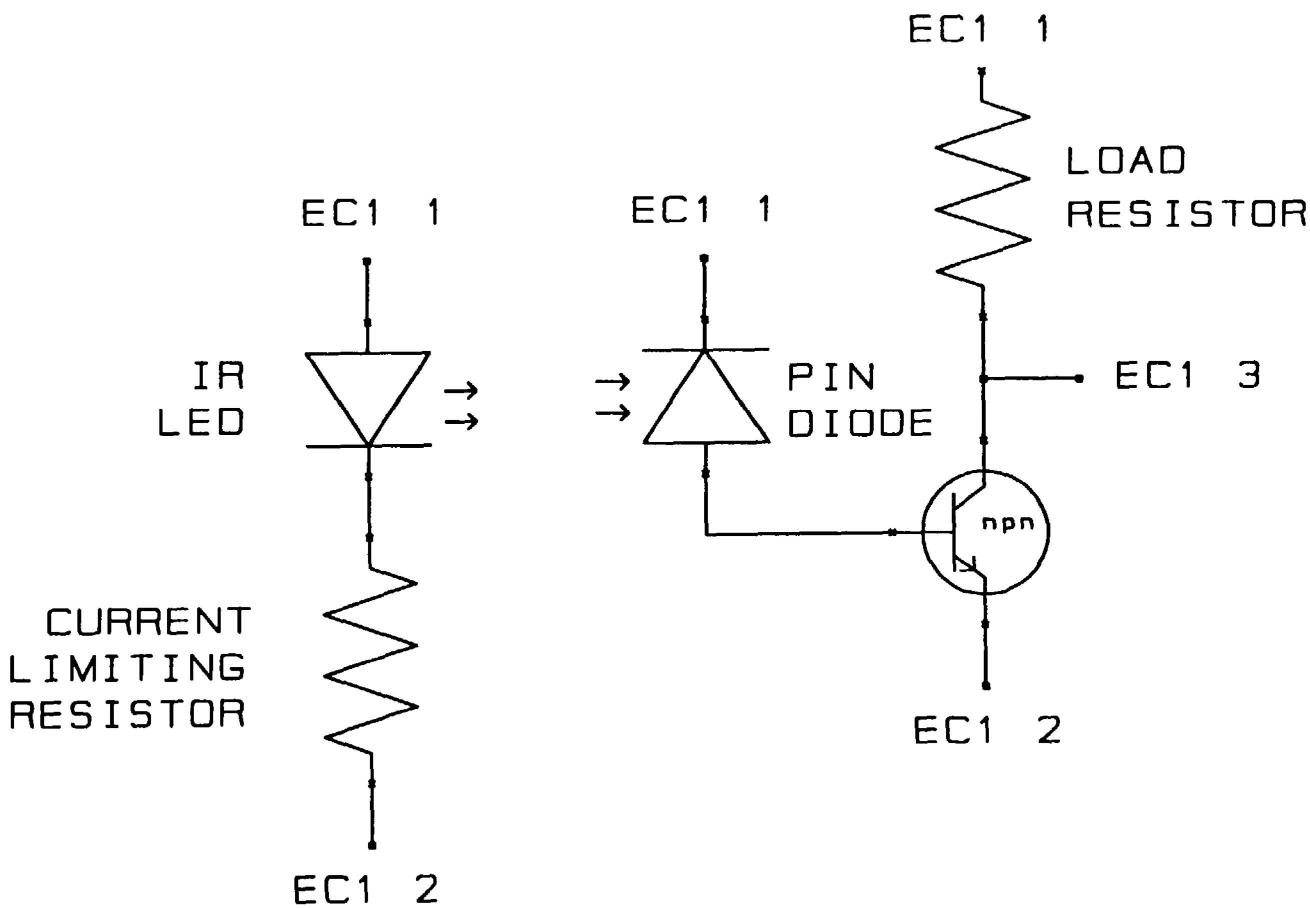


Figure 2. Electronic schematics of IR LED and PIN diode circuits.

times. This signal is further buffered by an IC inverter and input to the real-time counter clock (RTCC) pin of the microcontroller. As is discussed later, the RTCC feature of the 16C55 is used to tally the number of repetitive events.

This PIN diode signal processing scheme works well for scenarios where the PIN diode can be placed relatively close to the system's electronics. However, the long leads associated with remote placement of the photosensitive detector can introduce undesirable electronic noise. In such cases, it is prudent to include an additional stage of amplification at the PIN diode. The simple circuitry for such a preamplifier is illustrated in Figure 2.

2.3 Counting Interval Generation. The frequency of a repetitive event can be determined by tallying the number of cycles that occurs in a known interval of time. For additional flexibility, the event-counting time interval for this system is variable to allow the measurement of a fairly wide range of frequencies. This is achieved through the use of an LM555 timer which produces time intervals that can be adjusted by means of a potentiometer. The RTCC input of the 16C55 can be used as an 8-bit event counter. Therefore, as many as 255 ($2^8 = 256$) events can be counted in a single interval. The listed resistive and capacitive (RC) components, which determine the duration of the counting time interval, can vary this interval from less than 40 ms to almost 10 s. Thus, relatively high resolution frequency measurements can be obtained over the range of tens of hertz to well over 5 kHz.

This LM555 monostable timer operates under the control of the 16C55 microcontroller. The 16C55 signals the monostable timer that it is prepared to tally events through a *Count Ready* pulse that momentarily pulls the trigger input of the timer low. The timer responds by outputting a high *Count Window* pulse. It is the duration of this variable pulse that determines the counting interval. The microcontroller monitors this pulse and continues to tally events as long as it remains high.

2.4 Digital Frequency Display. A liquid crystal display (LCD) is included to show the measured frequency values. The display circuitry consists of a six-digit LCD module, a driver IC for each digit, and a fixed frequency oscillator that is required to make the enabled LCD segments visible. As discussed previously, the largest possible count value during any event measurement time interval is 255. Thus, the use of a six-digit display may appear to be a blatant example of overengineering. However, this display has been configured to allow recent frequency trends, as well as the current frequency value, to be visualized. As illustrated in Figure 3, three of the digits on the right of the display are configured for standard numeral output and are used to show the latest frequency measurement. The remaining three digits on the left of the display have been set up to show symbols rather than numerals. In particular, an upper square can be displayed, a lower square can be displayed, and a combination of both the upper and lower squares can be displayed. These symbols are used to indicate whether the measured frequency values are increasing in value, decreasing in value, or remaining constant.

The manner in which these digits can be configured to display either numerals or symbols is illustrated in Figure 4. Section A of this figure illustrates the conventional assignment of segment designations for a seven-segment digital display. The numeral 2 would be represented, as shown in section B of this figure, with the segments a, b, d, e, and g being enabled for display. This is the manner in which the three numeral digits of the six-digit display are designated and accessed. Section C of Figure 4 illustrates the segment designation for the digits that are configured to display symbols. If the data for the numeral 2 are sent to this symbol digit, the result will be as shown in section D--a square would be displayed in the upper portion of the digit. Thus, data for the number 2 appear as a "2" when routed to a numeral digit and as an upper square when routed to a symbol digit. Similarly, numeral 5 data will appear as a lower square, and numeral 8 data will appear as both upper and lower squares when sent to a symbol digit.

Each LCD digit is driven by a 74HC4543 binary-coded decimal (BCD) to 7-segment latch/decoder/driver IC. These ICs all receive BCD display data from a common 4-bit bus that is controlled by the 16C55. Inadvertent access of more than a

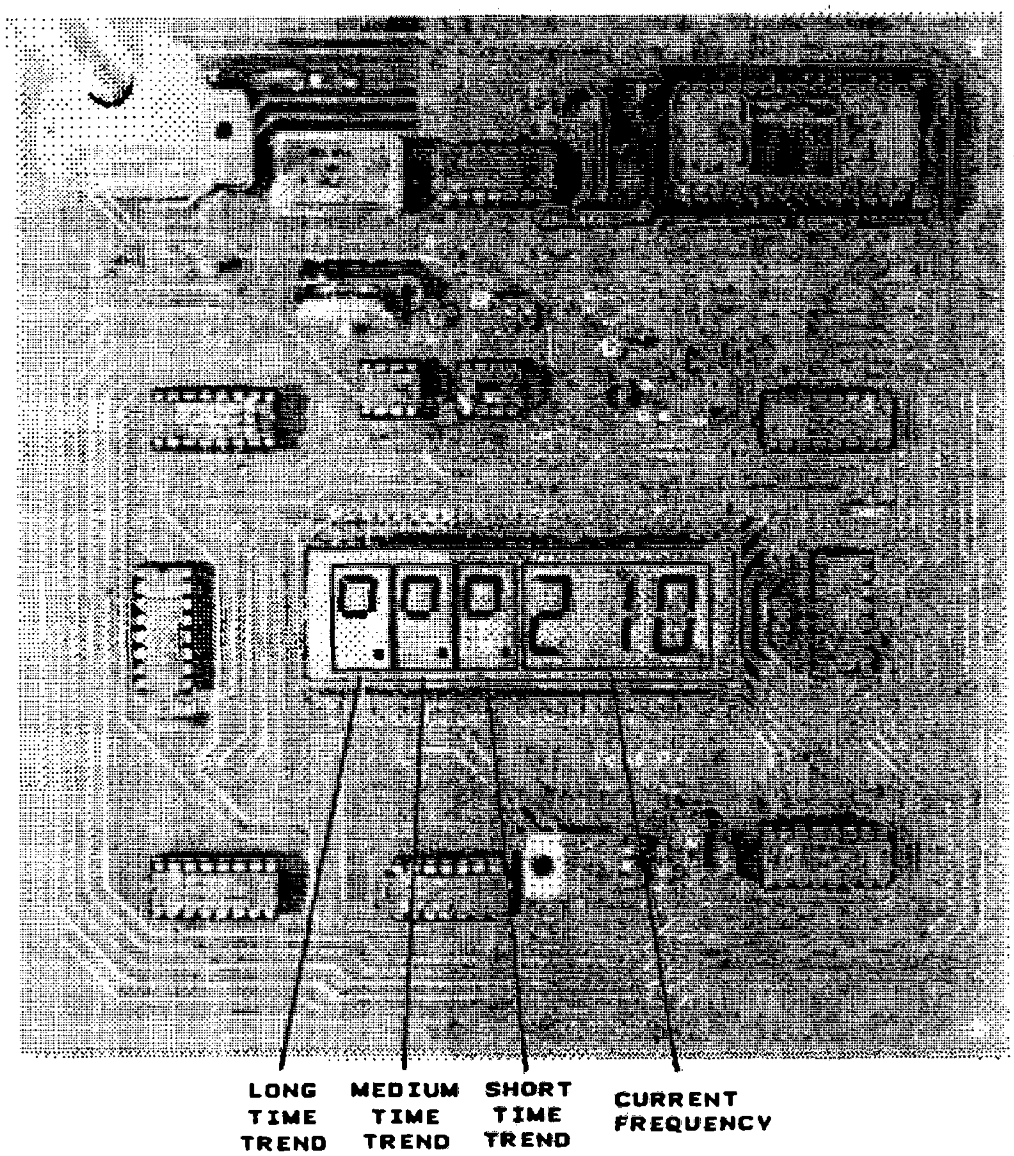


Figure 3. Illustration and explanation of frequency measurement digital display.

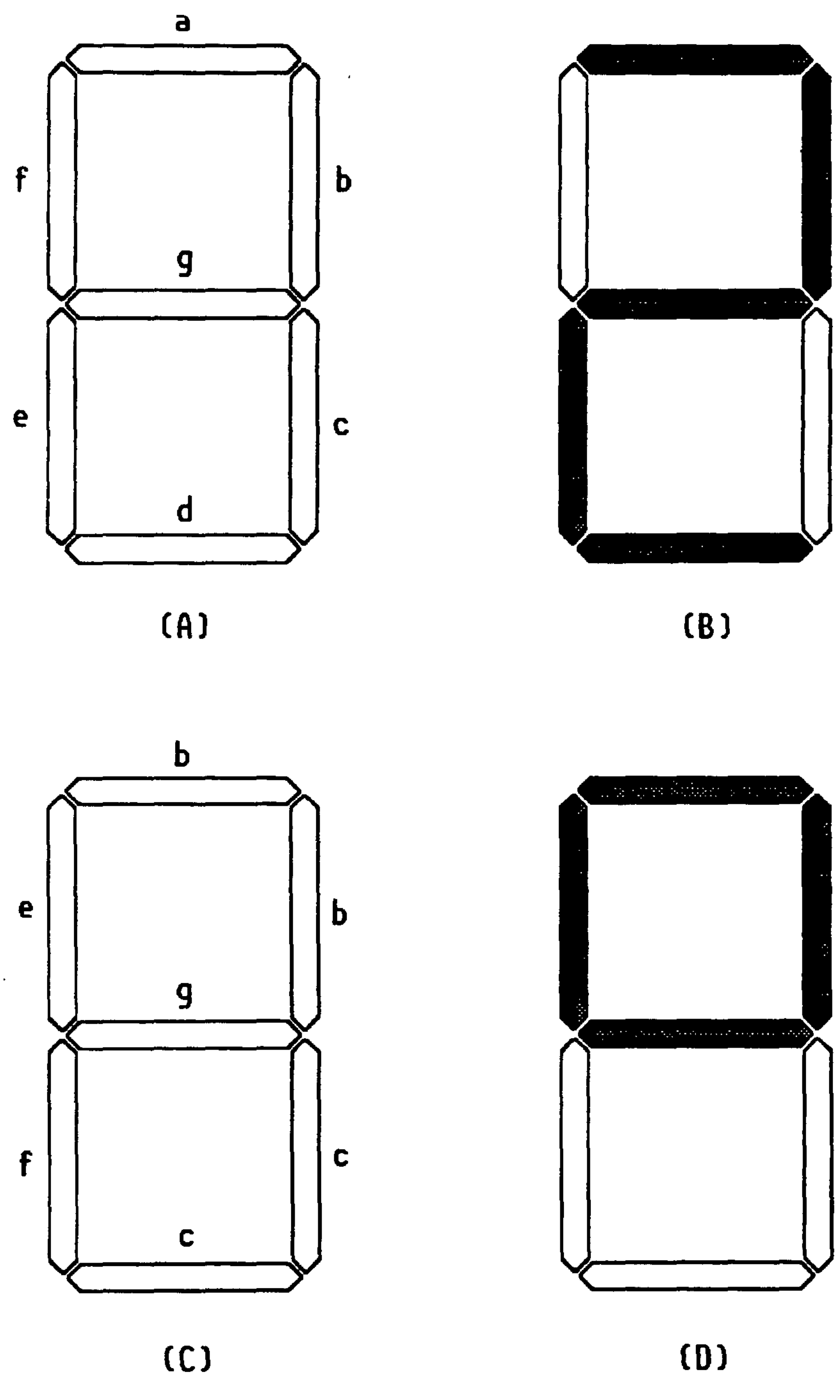


Figure 4. Designation of display segments for numeric and symbolic digits.

single 74HC4543 at a time is not a problem because these ICs do not latch data from the common bus until they are enabled by a momentary positive pulse to their latch disable (LD) pin. The LD pin of each 74HC4543 is routed to a separate 16C55 input/output (I/O) pin configured for output. Thus, the 16C55 transfers display data to a single LCD digit by first placing the desired data on the common display bus and then toggling the single LD pin that corresponds to the selected digit. Once the display data have been latched, the 74HC4543 decodes this BCD input to determine which segments of the chosen display digit are to be enabled for display. This method for driving the LCD display requires a 30-Hz to 200-Hz square wave signal that must be applied to the back-plane of the display and the phase input (PH) pins of the 74HC4543s. Such an LCD oscillator signal is conveniently provided by a second LM555 timer with RC components that provide a nominal 100-Hz signal.

A final feature of this digital display is the ability to display decimal points. The 16C55 controls the three decimal points associated with the symbolic display digits by accessing an equal number of tristate line drivers on a 74HC125. When accessed, each line driver will allow the corresponding decimal point to be displayed. If a line driver is not enabled, then its output assumes a high-impedance state and the decimal point is not visible. As is explained in more detail later, these decimal points are used to illustrate the system's mode of operation.

2.5 Microcontroller Interfacing. The 16C55 microcontroller orchestrates the frequency measuring and monitoring process through its multiple I/O lines. Consider first the frequency measurement process. The 16C55 announces its intention to make a frequency measurement by momentarily pulling its RA0 output line low. As previously discussed, this action initiates an LM555 timer that responds by generating a frequency counting interval pulse that is routed back to the 16C55 through the RA1 input pin. During the counting interval, the real-time clock/counter feature of the 16C55 is used to tally repetitive events. The RTCC operates in the following manner. Toggling the RTCC pin causes the microcontroller to automatically increase the value stored in an associated register by one. So, by clearing this register value when the frequency counting interval

is initiated, the RTCC register will automatically count the number of repetitive events. The 16C55 merely has to monitor the RA1 input to ascertain when the frequency counting interval comes to an end. At that time, the contents of the RTCC register is the number of events that occurred during the counting interval. This value is then transferred to a "scratch pad" register where the information is processed.

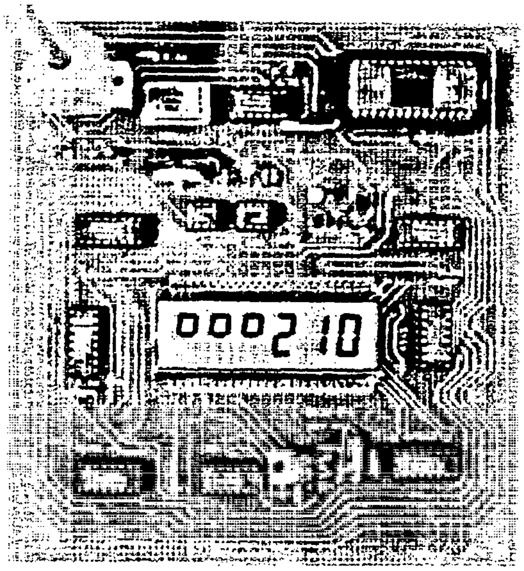
The RTCC counter operates in binary while we humans are more accustomed to dealing with decimal numbers. Therefore, microcontroller software converts the binary counter value to an equivalent three-digit decimal number that will be displayed on the LCD. This display is updated one digit at a time. Microcontroller output pins RB0, RB1, RB2, and RB3 are used to load the display data bus with the value of the digit. This information is simultaneously supplied to all the 74HC4543 LCD driver ICs. However, a data-latching pulse is only sent to the single driver IC for which the updated display data is intended. Latching pulses for the three numeric digits are supplied by microcontroller output pins RC0, RC1, and RC2 while the latching pulses for the three symbolic digits are furnished by output pins RC3, RC4, and RC5. Although the LCD is updated one digit at a time, this process occurs quickly enough that the entire display appears to change simultaneously.

The computing power of the 16C55 is used to monitor and display the recent trends in the frequency measurements in addition to the most recent measurement value. Internal registers are used to store the eight most recent frequency measurements. After every new measurement, average values are computed and compared for data spanning three different time intervals. First, the most recent frequency measurement value is compared to the previous measurement. This trivial computation provides trend information concerning the frequency variation over the shortest time interval. A frequency variation over a longer time interval is calculated by averaging the two most recent measurements and comparing that value to the average of the third and fourth most recent measurements. Finally, the frequency variation over the longest time interval is calculated by averaging the four most recent measurements and comparing that value to the average of the four measurements that preceded them. Results for each

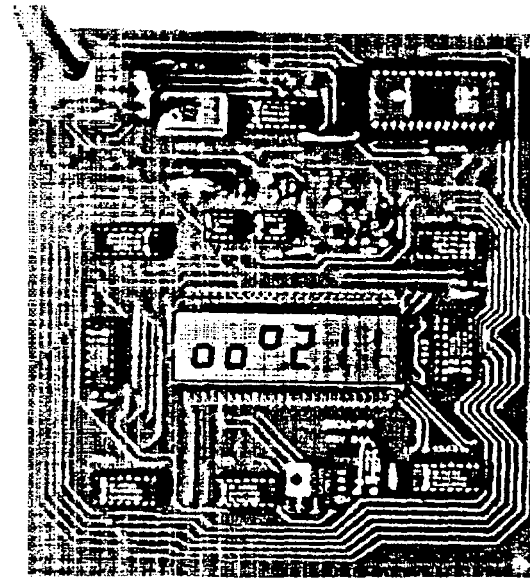
of these frequency trend calculations are displayed using the three symbolic digits, as exemplified in Figure 3. If the more recent average, over a given time interval, has increased in value, then an upper square is displayed. The lower square is displayed if a more recent time average has decreased in magnitude while both the upper and lower squares are displayed if an average value remains constant.

Figure 5 illustrates a number of display examples that depict a variety of scenarios. For each case, the three numerical digits on the right of the display show the most recently measured frequency value. In example A, the three symbolic digits on the left of the display are all showing upper squares. This indicates that the short-term, medium-term, and long-term frequency averages are all increasing. Clearly, the repetitive event that is being monitored is speeding up. Example B shows a long-term and medium-term decrease in frequency with a short-term increase in frequency. From this information it can be assumed that the repetitive process had been slowing down but has recently begun to speed up. Example C also shows a long-term frequency decrease and a short-term frequency increase. However, the medium-term symbolic display consists of both an upper and a lower square. Therefore, the repetitive event made a slower transition from the time that it was slowing down to the point where it began to speed up. In other words, the system being monitored maintained a relatively constant frequency at the frequency minimum. Finally, the display in example D indicates that for the longer time averages, the frequency has been constant. Knowledge of the system's frequency response would be required to decide whether the increase in the short-term frequency was due to an actual change in frequency or random noise.

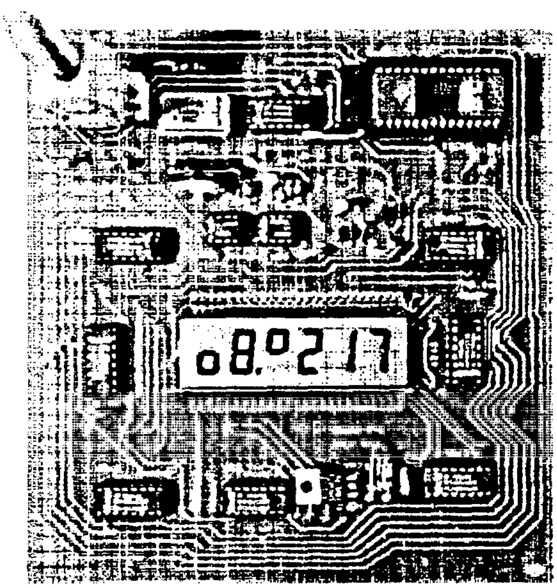
Measuring and displaying frequency trend information that spans a variety of integration times allow the user to selectively focus on the information that is most suited to the specific application. Short-term variations may represent little more than noise in repetitive systems that are not capable of rapid frequency variations. For these scenarios, the long-term trend measurements can average out the short-term noise and provide more useful information. Conversely, the short-term trend data may be more



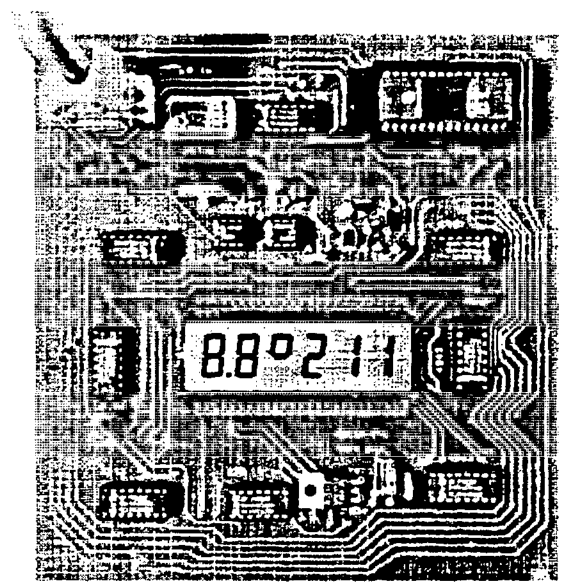
(A)



(B)



(C)



(D)

Figure 5. Examples of frequency measurement digital displays.

applicable to systems that are capable of rapid frequency response or where short-term variations in frequency are particularly critical.

Sitting and staring at a digital display is probably not a work assignment that a great many technical personnel would fight over. Some of the remaining I/O pins on the 16C55 can be used to avoid such an unpleasant task. Three of these pins, RA2, RA3, and RB7, have been configured as output pins and are available to drive external digital devices through connector pins EC1 7, EC1 8, and EC1 9. One particularly useful and simple external device is the piezoelectric buzzer. Under software control, such buzzers can be used to alert the user of frequency conditions that are outside a previously programmed range of acceptable operating conditions. The availability of multiple digital outputs allows the use of buzzers of different tones so that the nature of the unacceptable condition can be readily determined from the sound of the alarm. When the repetitive system is operating as required, the user can be rewarded with silence and will be able to concentrate on other matters.

These digital outputs can also be used to actively control the rate of a repetitive event. Thus, in addition to frequency measurement and monitoring, a feedback system that is used to dynamically regulate the frequency of the repetitive event can be incorporated. As an example, consider the case of the optical chopper with its rotating slotted wheel. The digital output pins can be used to increase or decrease the value of a digital counter depending on whether the measured frequency is determined to be too high or too low. Such a counter can drive a digital-to-analog converter (DAC) that controls the voltage that is supplied to the optical chopper motor. Thus, variations in measured rotational frequencies from a desired value can be used to alter the speed of the chopper motor in a manner that will decrease the frequency error. Depending on the required level of sophistication, proportional-integral-differential (PID) algorithms are available for precise motor control (Frank 1992).

The decision-making process for controlling these alarms and feedback loops can be based on either the most recent frequency measurement or one of the computed

frequency trend measurements. Recall that there are three different time intervals over which frequency trends are calculated. Therefore, some means must be provided for allowing the user to choose a specific trend interval and informing the user of which frequency trend interval is currently being used to make alarm and feedback decisions. Two 16C55 pins, RC6 and RC7, which are accessable through connector pins EC1 4 and EC 1 5, have been configured as digital inputs and are available for user control. One of these input lines can be used to sequence through the three time intervals of frequency trend calculation for digital output control. The presently selected time interval is denoted by enabling the decimal point that is associated with the symbol digit that corresponds to the chosen trend time interval. These decimal points are accessed through the output lines RB4, RB5, and RB6 of the 16C55. Figure 5 illustrates the decimal point displays associated with the four possible alarm decision modes. In example A, all the decimal points are turned off, which indicates that the alarm feature has been disabled or is operating from the most recent frequency measurement. Examples B, C, and D each have a single decimal point enabled to show that short-term, medium-term, and long-term averages are respectively being used to control the alarm and feedback functions.

A natural consequence of a world littered with microwave ovens, telephone answering machines, digital watches, radar detectors, and other overly vocal devices is the evolution of man's ability to totally ignore audio warning devices of all types. Those of us who have developed a particularly acute case of selective deafness require vast amounts of audio stimulation to get our attention. Individuals who are still conscious of their surroundings require much less prompting and are in fact prone to irritation when exposed to the stimulation levels that many of us require. In an attempt to satisfy all and to have technology work for the individual, this frequency measurement system has been configured to emit warning signals of user-selectable duration. With appropriate programming, one of the two digital input lines of the 16C55 can be used to toggle through a series of alarm duration settings. In this way, the length of the alarm signals can be selected to match the user's state of awareness.

The operation of the 16C55 microcontroller is paced by a crystal oscillator signal that is input through pin OCS1. This hardware system has been operated at oscillator clocking frequencies as high as 10 MHz. However, in practice a 1-MHz clocking frequency is sufficiently fast and avoids the need for overly long software-controlled delays that are required by some of the external hardware components. Finally, in the unlikely event that the 16C55 becomes confused, the entire microcontroller can be reset by momentarily grounding the MCLR master clear input line that is accessable through connector pin EC1 6.

Appendix A contains information that will be useful to individuals who wish to duplicate this hardware. In particular, this appendix includes a parts list, printed circuit board masks, and a table which lists the connector pin signals.

3. SYSTEM SOFTWARE

The 16C55 microcontroller contains 512 words of EPROM that are available for the long-term storage of driver software. These programs can be erased and changed at will but are nonvolatile. That is, software that is stored in the 16C55 is not lost when the power is turned off. Programs are written in 16C55 machine language, which makes some sort of an assembler-lister-compiler a virtual necessity. Microchip Technology Inc. offers a PICSTART-16B microcontroller development system that includes PC-compatible assembler software, simulator software, programmer software, and a programmer board. The MPALC macro assembler is a symbolic cross assembler that converts human-generated source code into object code for the entire family of PIC16CXX microcontrollers. Use of a macro assembler allows frequently accessed portions of the source code to be written as subroutine-like segments that can then be conveniently called using a single command. Programs can be tested and debugged using the MPSIM* software simulator before being loaded into the microcontroller using the

* PICSTART, MPALC, MPSIM, and PIC are trademarks of Microchip Technology Inc.

programming software and board. An ultraviolet EPROM eraser is used to clear the microcontrollers' program memory when the driving software needs to be modified.

A flowchart of the program that is used to monitor chopper wheel rotational frequencies is illustrated in Figure 6. In this flowchart, the names of macros that are used to accomplish various tasks are contained in square brackets. A listing of the source code for this program is included in Appendix B.

When powered up or following a master reset, the 16C55 microcontroller resets the program counter register to contain all "1"s. The first operational clock cycle will roll the program counter over to contain all "0"s. Therefore, a reset condition effectively begins instruction processing from a program memory address of zero. These initial instructions are programmed to define the microcontroller's I/O port directions. Each I/O pin on the 16C55 can serve as either an input or an output data pin. The default reset value is to have all the I/O pins set as high-impedance inputs. Various pins are manually defined as outputs so that the microcontroller can send data to the various display digits, alarms, and feedback loops. After initial default values are defined for the alarm duration and all the decimal point displays are turned off, the I/O port directions are once again defined. During this step, the port directions are in fact defined to the exact same configuration as they were at the beginning of the program. This apparently redundant process is included because it serves as the start of the iterative frequency measurement loop. The manufacturer notes that when operated in noisy environments, the I/O control register, which stores the defined I/O directions, can become corrupted. This potentially problematic condition is skirted by redefining the I/O control register at the beginning of every frequency measurement operation.

At this point in time, the microcontroller polls the digital input line that the user accesses to request a change in the duration of the alarm signal. If accessed, the value of the alarm duration register is incremented to the next available value. There are eight different alarm durations ranging from short clicks to obnoxiously long blasts. Each variation request implements the next longer alarm duration until the longest value is

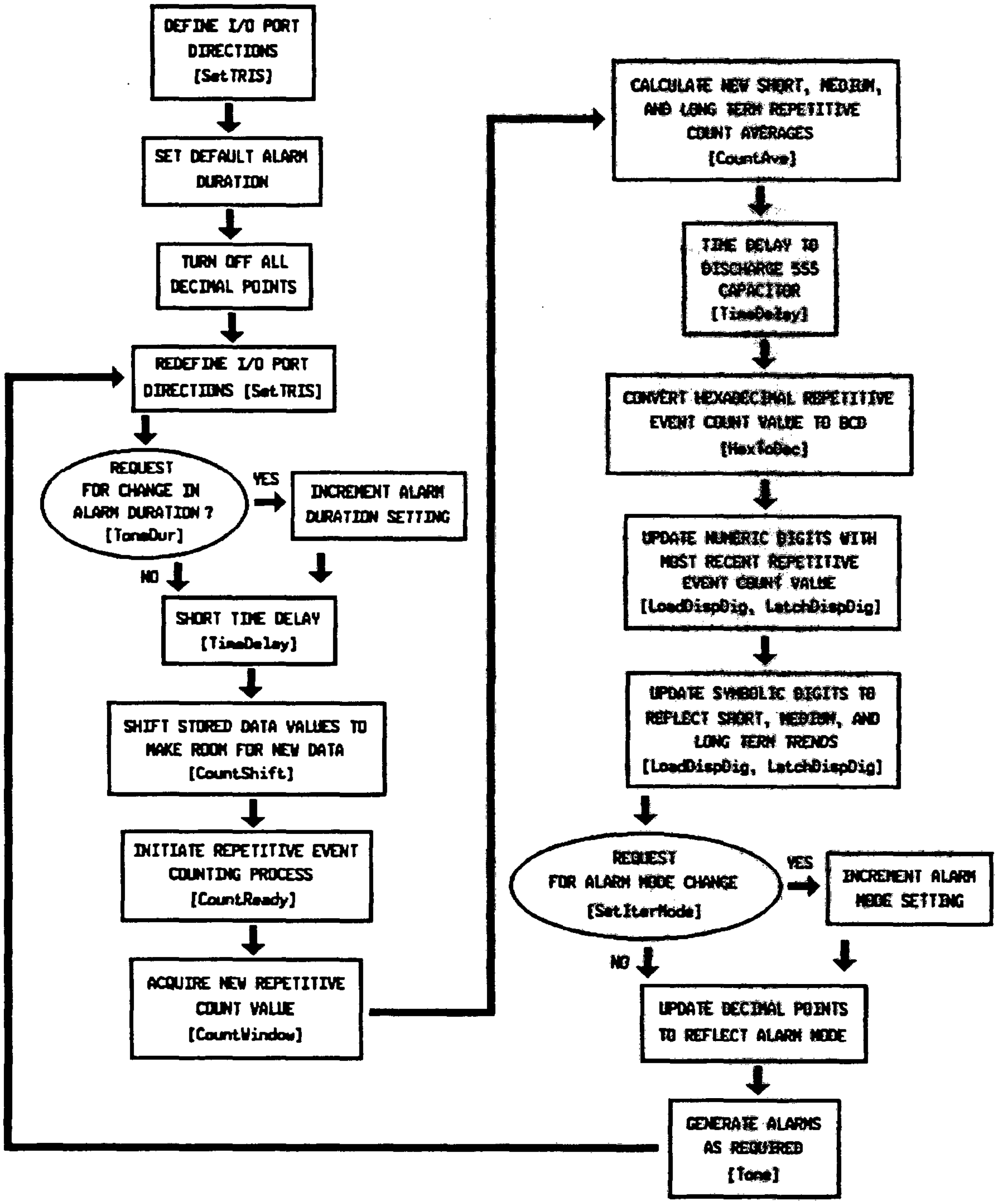


Figure 6. Flowchart of frequency measurement microcontroller software.

used. The next variation request cycles back to the shortest duration value. If a request is not being made for a change in alarm duration, then no action is taken other than the implementation of a short time delay to allow all internal and external devices to settle.

The eight most recent frequency data values are stored so that recent trends in the frequency measurements can be calculated and displayed. These values are stored in eight microcontroller registers that serve as a serial buffer. To make room for the next frequency data value, each resident value is shifted to an adjacent register. The effect of this process is to lose the oldest data value while leaving a space available for the next frequency measurement value.

The microcontroller initiates the frequency measurement counting process by outputting a negative pulse to the trigger input of an LM555. This action starts a user-adjustable time delay that defines the time interval over which repetitive events will be tallied. The microcontroller clears the RTCC register that will automatically track the repetitive events and then monitors the LM555 output for an indication that the counting interval has terminated. When the counting interval ends, the count value in the RTCC register is transferred to the first register in the serial buffer, which was vacated by the previous data-shifting process. Averages are then computed for the data stored in the serial buffer registers so that frequency trends over three time intervals can be determined and displayed.

An LM555 generates well-defined time intervals by monitoring the amount of time required for an external capacitor in an RC circuit to charge to some threshold voltage. At the end of the time interval, this capacitor is discharged and the LM555 is ready to generate another time interval. The operation of the 16C55 microcontroller is rapid enough that subsequent frequency measurement time intervals can be initiated before the LM555 timing capacitor has had a chance to fully discharge. This undesirable condition is avoided by including a microcontroller-generated time delay that allows the capacitor in the LM555 timing circuit to fully discharge prior to the start of the next measurement time interval.

The RTCC register that tallies the repetitive events is an 8-bit binary register. Therefore, the final count value is contained in this register as a series of "0"s and "1"s. These 8 bits could be divided into two 4-bit nibbles and visually displayed as two hexadecimal digits. However, hexadecimal numbers, such as D8h and 4Bh, are not as meaningful to 10-fingered humans as their decimal equivalents, 216 and 75. So, the frequency measurements are converted by software within the microcontroller to decimal values. The converted decimal digits are then displayed along with the symbolic digits that indicate the frequency trends over three different time intervals.

At this point, the microcontroller polls the digital input line that the operator accesses to request a change in the alarm or feedback mode. For the sample software in Appendix B, there are four possible alarm modes. The first of these modes is a state where all alarm functions are disabled. The remaining modes access the alarm for frequency reductions as determined by either short-term, medium-term, and long-term averaging. A request for an alarm change enables the next available alarm mode. Therefore, in order to change to a specific alarm mode, a number of mode requests may be required until the desired alarm function is obtained. The decimal points associated with the symbolic display digits are then updated to reflect the current alarm mode. Finally, a digital output line is accessed to sound an alarm as required. At this time, a single cycle of frequency measurement and display has been completed, and the software loops back to repeat the entire cycle.

4. SUMMARY

An integrated hardware/software system that was developed to economically measure, monitor, and control the frequency of repetitive events has been presented. A user-adjustable counting interval allows high-resolution frequency measurements to be obtained for repetitive events with frequencies that range from tens of hertz to several kilohertz. The flexibility of this system is further enhanced by the inclusion of microprocessor control that allows both frequency values and trends to be determined and

displayed. An assortment of digital inputs and outputs can be used to interface this system with external devices such as alarms and frequency feedback controllers. Schematic diagrams, circuit descriptions, printed circuit board masks, and a parts lists of the hardware are included. Sample software that can be used to drive the microprocessor is also presented and discussed.

Intentionally Left Blank

5. REFERENCES

Frank, S. "Intelligent Remote Positioner (Motor Control) Using the PIC16C5X." Application note AN531 from Embedded Control Handbook, Microchip Technology, Chandler, AZ, 1992.

Microchip Technology Inc. Microchip Data Book. Second Edition, Chandler, AZ, 1992.

Intentionally Left Blank

APPENDIX A:

HARDWARE FABRICATION INFORMATION

Intentionally Left Blank

This appendix contains information that is useful for the fabrication of the hardware that is presented in this report. In particular, this section contains a parts list, printed circuit board masks, and a table which lists the connector signals.

Table A-1. Frequency Measurement Hardware Parts List

DESCRIPTION	QUAN	PART NUMBER *	COST
8-bit microcontroller, 16C55	1	PIC16C55/JW-ND	19.13
quartz crystal oscillator, 1 megahertz	1	X101-ND	4.88
9-pin D sub connector, male	1	A2043-ND	1.39
9-pin D sub connector, female	1	A2047-ND	1.45
74HC04 hex inverter	1	MM74HC04N-ND	0.28
74HC125 tri-state quad buffer	1	MM74HC125N-ND	0.53
74HC4543 BCD-to-7 segment decoder/driver	6	MM74HC4543N-ND	9.00
LM555 timer	2	MN555CN-ND	3.30
LM340-5 5-volt regulator, TO-220 package	1	LM340AT-5.0-ND	2.25
2N3904 npn transistor, TO-92 package	1	2N3904-ND	0.31
2N3906 pnp transistor, TO-93 package	1	3N3906-ND	0.31
6-digit liquid crystal display	1	LCD005VT-ND	7.74
potentiometer, 100-kilohm	1	U201R104B-ND	0.43
capacitor, 100-microfarad, 25-volt	1	P6239	0.13
capacitor, 10-microfarad, 12-volt	1	P6343	0.22
capacitor, 0.1-microfarad, ceramic	6	P4910	2.58
capacitor, 0.01-microfarad, ceramic	2	P4904	0.42
resistor, 1-kilohm, 1/4 watt	1	1.0KQ	0.26
resistor, 4.7-kilohm, 1/4 watt	4	4.7KQ	0.26
resistor, 10-kilohm, 1/4 watt	1	10KQ	0.26
resistor, 20-kilohm, 1/4 watt	1	20KQ	0.26
resistor, 100-kilohm, 1/4 watt	2	100KQ	0.26
socket, zero insertion force, 28-pin	1	A303-ND	10.48
socket, 8-pin dual inline parallel	2	A9408-ND	0.80
socket, 14-pin dual inline parallel	3	A9414-ND	1.98
socket, 16-pin dual inline parallel	6	A9416-ND	4.50
printed circuit board (cost can vary)	1		100.00
		total cost of hardware:	173.41

* Part numbers are as listed in Digi-Key catalog 941. Digi-Key Corporation, P.O. Box 667, Thief River Falls, MN 56701; Tel: (800) 344-4539; FAX: (218) 681-3380

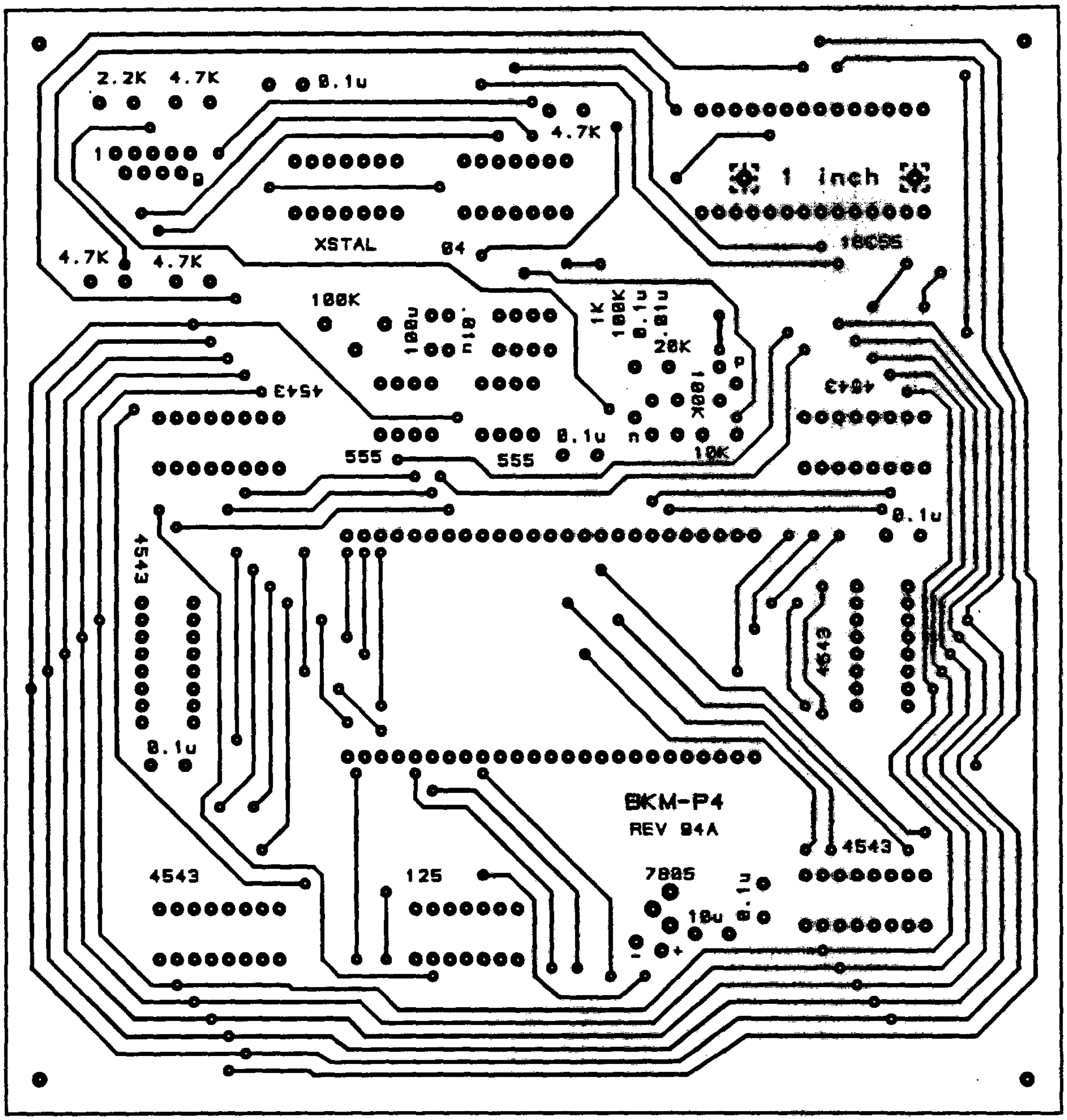


Figure A-1. Printed circuit board mask of component side.

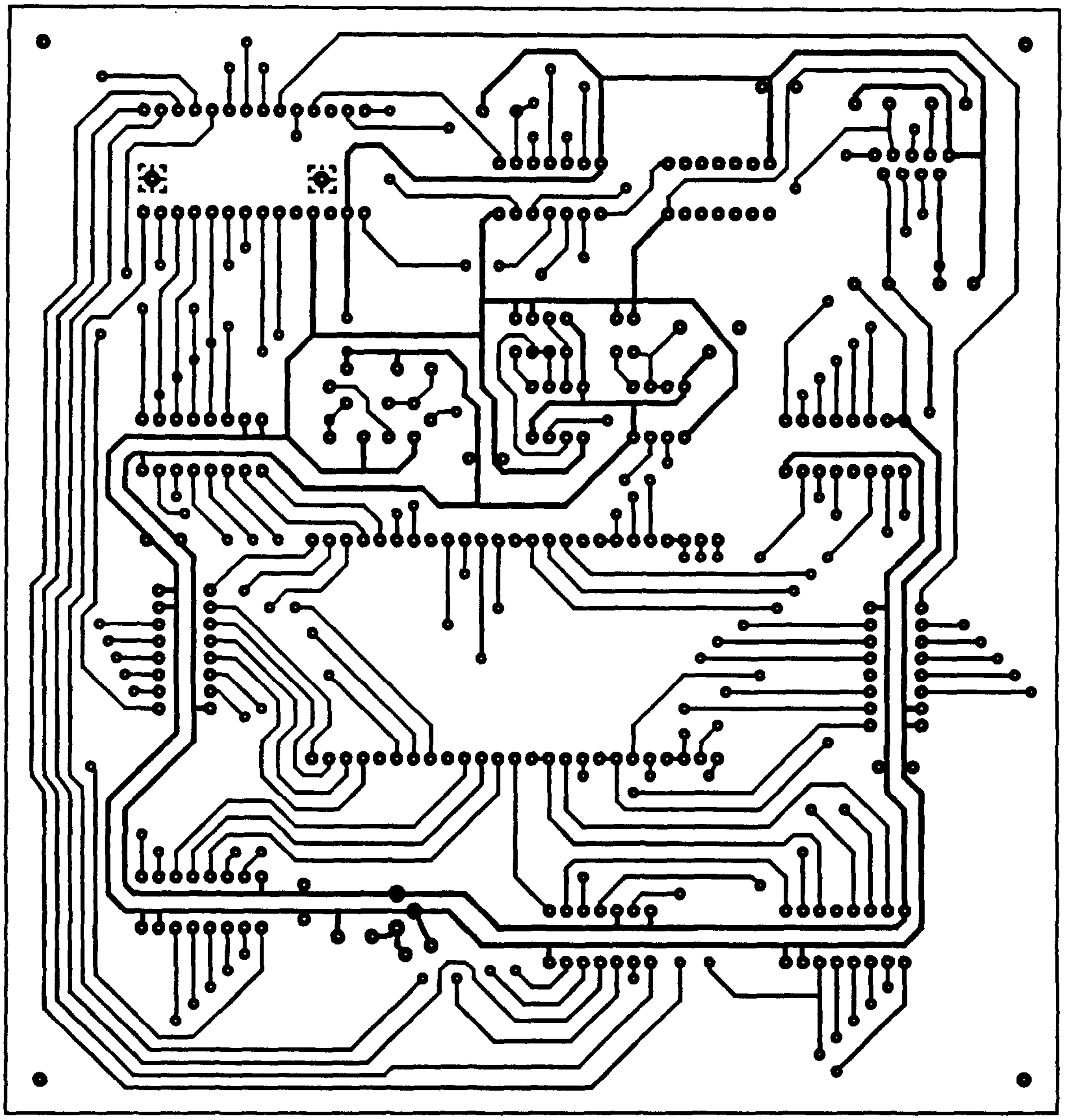


Figure A-2. Printed circuit board mask of noncomponent side.

Table A-2. Connector Pin Signals

Pin No.	Signal Description
EC1 1	+5 volts
EC1 2	ground
EC1 3	pin diode input
EC1 4	digital input, IA
EC1 5	digital input, IB
EC1 6	master reset
EC1 7	digital output, OA
EC1 8	digital output, OB
EC1 9	digital output, OC

APPENDIX B:

DRIVER SOFTWARE LISTING

Intentionally Left Blank

This appendix contains the software that was used to monitor optical chopper wheel rotation frequencies. The code is written for the Microchip 16C55 microcontroller using the MPALC macro assembler. Tables which enumerate and briefly describe the macros and register files along with listings of the macros and the main program are provided.

```
; DRIVER SOFTWARE FOR FREQUENCY MONITORING HARDWARE

;***************************** MACRO DIRECTORY ******************************

;   MACRO       PARAMETER
;   NAME          NAME                        DESCRIPTION
; ----------  --------- ----------------------------------------------------
; SetTRIS                   Initializes the bidirectional port I/O directions.
; TimeDelay     Length      Generates a time delay with duration determined by
;                           the parameter magnitude. F08 and F1C are utilized as
;                           counting registers.
; CountReady                Generates a negative count ready pulse. Calls
;                           TimeDelay.
; CountWindow               Monitors the number of RTCC pulses during the positive
;                           count window pulse. Final RTCC value stored in F09.
; HexToDec                  Converts the Hexidecimal value in F09 to BCD values in
;                           F0A (units), F0B (tens) and F0C (hundreds).
; LoadDispDig               Loads the display data bus with value preloaded in
;                           F08. Does not disturb the state of the decimal points.
; LatchDispDig  DigPos      Generates a positive Latch Disable pulse which latches
;                           the display data for the specified digit. 0=units,
;                           1=tens, 2=hunds, 3=SI, 4=MI, 5=LI. Calls TimeDelay.
; CountShift                Transfers previously measured count values to next
;                           higher file location. F10->F11, F11->F12,....F16->F17.
; CountAve                  Calculates the count averages for integrated
;                           comparison. Uses F18 through F1B.
; SetIterMode               Determines which integration mode drives speaker. Uses
;                           F1D.
; Tone                      Generates a high pulse on 0A which can be used to
;                           power a piezoelectric buzzer.
; ToneDur                   Varies the duration of the tone.
;
;****************************************************************************
;****************************************************************************
;
; FILE USAGE
;
; F08: scratch pad register.
; F09: decrementing register for Hex to BCD conversion.
```

```
; F0A: BCD units value.
; F0B: BCD tens value.
; F0C: BCD hundreds value.
; F0D:
; F0E: average of count values N-4 and N-5
; F0F: average of count values N-6 and N-7
; F10: count value N
; F11: count value N-1
; F12: count value N-2
; F13: count value N-3
; F14: count value N-4
; F15: count value N-5
; F16: count value N-6
; F17: count value N-7
; F18: average of count values N and N-1
; F19: average of count values N-2 and N-3
; F1A: average of count values N, N-1, N-2 and N-3
; F1B: average of count values N-4, N-5, N-6 and N-7
; F1C: time delay most significant decriment counter
; F1D: stores integration mode speaker mode
; F1E: stores speaker tone duration value
;
;******************************************************************************

; MACRO FOR INITIALIZING THE PORT I/O DIRECTIONS --------------------------

SetTRIS         MACRO                           ;start of MACRO to set TRIS
                                                ;0 = output, 1 = input
                MOVLW           B'0010'         ;port A I/O direction template
                TRIS            5               ;load port A TRIS register

                MOVLW           B'00000000'     ;port B I/O direction template
                TRIS            6               ;load port B TRIS register

                MOVLW           B'11000000'     ;port C I/O direction template
                TRIS            7               ;load port C TRIS register

                ENDM                            ;end of MACRO

; MACRO FOR GENERATING A TIME DELAY ---------------------------------------

TimeDelay       MACRO           Length          ;start of MACRO for time delay
                                                ;Length determines delay
                                                ;duration

                LOCAL           InTimeLoop      ;define local labels
                LOCAL           OutTimeLoop     ;

                MOVLW           Length          ;move time delay length value
                MOVWF           0x1C            ;into F1C...
```

```
OutTimeLoop     MOVWF       0x08            ;and F08
InTimeLoop      DECFSZ      0x08,1          ;decriment F08
                GOTO        InTimeLoop      ;if not zero go to InTimeLoop
                CLRWDT                      ;clear watch dog timer
                DECFSZ      0x1C,1          ;decriment F1C
                GOTO        OutTimeLoop     ;if not zero go to OutTimeLoop

                ENDM                        ;end of MACRO

; MACRO FOR GENERATING A NEGATIVE COUNT READY PULSE -------------------------

CountReady      MACRO                       ;start of count ready MACRO

                BCF         5,0             ;drop the count ready line
Length          SET         0x02
                TimeDelay   Length          ;time delay to allow settling
                BSF         5,0             ;raise the count ready line

                ENDM                        ;end of MACRO

; MACRO FOR COUNTING THE NUMBER OF RTCC PULSES ------------------------------


CountWindow     MACRO                       ;start of count window MACRO

                LOCAL       CntWinLoop      ;define local label

                MOVLW       B'00101111'     ;load W with OPTION parameters

                OPTION                      ;transfer W to OPTION register

                CLRF        1               ;clear F01, the RTCC counter

CntWinLoop      CLRWDT                      ;clear watch dog timer

                BTFSC       5,1             ;has count window closed ???
                GOTO        CntWinLoop      ;(no)
                MOVF        1,0             ;(yes) move F01 (RTCC value)
                                            ;to W
                MOVWF       0x09            ;store final RTCC value in F09

                MOVWF       0x10            ;store final RTCC value in F10

                ENDM                        ;end of MACRO

; MACRO FOR CONVERTING HEX DATA TO BCD DATA ---------------------------------

HexToDec        MACRO                       ;start of hex to dec MACRO
                                            ;hex data preloaded in F09

                LOCAL       HexDecMore      ;define local labels
```

```
            LOCAL       HexDecUnit      ;
            LOCAL       HexDecTens      ;
            LOCAL       HexDecHund      ;
            LOCAL       HexDecDone      ;
            LOCAL       TensNotBlank    ;
            LOCAL       HundNotBlank    ;

            MOVLW       B'00000001'     ;load W with the value "1"
            MOVWF       0x0A            ;trans "1" to BCD units file
            MOVLW       B'00001010'     ;load W with the value "10"
            MOVWF       0x0B            ;trans "10" to BCD tens file
            MOVWF       0x0C            ;trans "10" to BCD hunds file

            MOVF        0x09,0          ;transfer F09 to W
            BTFSS       3,2             ;zero bit set ???
            GOTO        HexDecMore      ;(no) go to HexDecMore
            MOVLW       B'00000000'     ;load W with the value "15"
            MOVWF       0x0A            ;trans "15" to BCD units file
            GOTO        HexDecDone      ;leave routine w/ "blanks"

HexDecMore  DECFSZ      0x09,1          ;dec F09, is result zero ???
            GOTO        HexDecUnit      ;(no)
            GOTO        HexDecDone      ;(yes)
HexDecUnit  INCF        0x0A,1          ;increment units value
            MOVLW       0xA             ;load W with the value "10"
            SUBWF       0x0A,0          ;subtract "10" from unit value
            BTFSS       3,2             ;zero bit set ???
            GOTO        HexDecMore      ;(no) jump to next decriment
HexDecTens  CLRF        0x0A            ;(yes) load "0" into unit file
            INCF        0x0B,1          ;increment tens value
            MOVLW       0x0B            ;load W with "11"
            SUBWF       0x0B,0          ;subtract W from F0B
            BTFSS       3,2             ;zero bit set
            GOTO        TensNotBlank    ;(no) go to TensNotBlank
            MOVLW       B'00000001'     ;(yes) load W with "1"
            MOVWF       0x0B            ;transfer W to F0B
TensNotBlank MOVLW      0xA             ;(no) load W with value "10"
            SUBWF       0x0B,0          ;subtract "10" from tens value
            BTFSS       3,2             ;zero bit set ???
            GOTO        HexDecMore      ;(no) jump to next decriment
HexDecHund  CLRF        0x0B            ;(yes) load "0" into tens file
            INCF        0x0C,1          ;increment hundreds file
            MOVLW       0x0B            ;load W with "11"
            SUBWF       0x0C,0          ;subtract W from F0C
            BTFSS       3,2             ;zero bit set ???
            GOTO        HundNotBlank    ;(no) go to HundNotBlank
            MOVLW       B'00000001'     ;(yes) load W with "1"
            MOVWF       0x0C            ;transfer W to F0C
HundNotBlank GOTO       HexDecMore      ;jump to next decriment

HexDecDone  NOP
```

```
                ENDM                                  ;end of MACRO

; MACRO FOR LOADING DISPLAY DIGIT DATA INTO RB ------------------------------

LoadDispDig     MACRO                                 ;start of digit loading MACRO
                                                      ;digit value is assumed to
                                                      ;already be in F08

                MOVLW           B'11110000'           ;load AND mask into W
                ANDWF           6,1                   ;clear lower nibble of RB
                MOVF            8,0                   ;load W with contents of F08
                IORWF           6,1                   ;combine digit value and upper
                                                      ;nibble of RB, leave in RB
                ENDM

; MACRO FOR LATCHING DISPLAY DIGIT VALUE ------------------------------------

LatchDispDig    MACRO           DigPos                ;start of unit latch MACRO

Length          SET             0x02                  ;set the value of Length

                BSF             7,DigPos              ;digit latch bit high
                TimeDelay       Length                ;time delay for settling
                BCF             7,DigPos              ;digit latch bit low

                ENDM                                  ;end of MACRO

; MACRO FOR SHIFTING COUNT VALUES IN FILES ----------------------------------
;
CountShift      MACRO                                 ;start of count shift MACRO

                LOCAL           ShiftAgain            ;define local label

                MOVLW           0x07                  ;load W with "7"
                MOVWF           0x08                  ;transfer W to F08

                MOVLW           0x17                  ;load W with "17"
                MOVWF           0x04                  ;transfer W to F04

ShiftAgain      DECF            0x04,1                ;decriment FSR by one

                MOVF            0x00,0                ;load W with contents of file
                                                      ;designated by F04
                INCF            0x04,1                ;increment FSR by one
                MOVWF           0x00                  ;transfer W to file designated
                                                      ;by F04
                DECF            0x04,1                ;decriment FSR by one

                DECFSZ          0x08,1                ;decriment F08, skip if "0"
                GOTO            ShiftAgain            ;go to again
```

```
            ENDM                            ;end of MACRO

; MACRO FOR CALCULATING COUNT AVERAGES ------------------------------------

CountAve    MACRO                           ;start of count average MACRO

            MOVF        0x10,0              ;load W with F10
            ADDWF       0x11,0              ;add F11 to W, store in W
            MOVWF       0x18                ;transfer W to F18
            RRF         0x18,1              ;rotate F18 one bit to right

            MOVF        0x12,0              ;load W with F12
            ADDWF       0x13,0              ;add F13 to W, store in W
            MOVWF       0x19                ;transfer W to F19
            RRF         0x19,1              ;rotate F19 one bit to right

            MOVF        0x18,0              ;load W with F18
            ADDWF       0x19,0              ;add F19 to W, store in W
            MOVWF       0x1A                ;transfer W to F1A
            RRF         0x1A,1              ;rotate F1A one bit to right

            MOVF        0x14,0              ;load W with F14
            ADDWF       0x15,0              ;add F15 to W, store in W
            MOVWF       0x0E                ;transfer W to F0E
            RRF         0x0E,1              ;rotate F0E one bit to right

            MOVF        0x16,0              ;load W with F16
            ADDWF       0x17,0              ;add F17 to W, store in W
            MOVWF       0x0F                ;transfer W to F0F
            RRF         0x0F,1              ;rotate F0F one bit to right

            MOVF        0x0E,0              ;load W with F0E
            ADDWF       0x0F,0              ;add F0F to W, store in W
            MOVWF       0x1B                ;transfer W to F1B
            RRF         0x1B,1              ;rotate F1B one bit to right

            ENDM                            ;end of MACRO

; MACRO FOR DETERMINING WHICH ITERATION MODE DRIVES SPEAKER -----------------
;
SetIterMode MACRO                           ;start of MACRO

            LOCAL       SetIterEnd          ;define local variables
            LOCAL       SetIterSI           ;
            LOCAL       SetIterMI           ;
            LOCAL       SetIterLI           ;
            LOCAL       SetIterNo           ;

            BTFSS       0x07,6              ;is IA activated ???
            GOTO        SetIterEnd          ;(no) mode stays the same
```

```
                BTFSC       0x1D,0          ;(yes) first bit of F1D set???
                GOTO        SetIterMI       ;(yes) go to SetIterMI
                BTFSC       0x1D,1          ;(no) second bit of F1D set???
                GOTO        SetIterLI       ;(yes) go to SetIterLI
                BTFSC       0x1D,2          ;(no) third bit of F1D set???
                GOTO        SetIterNo       ;(yes) go to SetIterNo
                GOTO        SetIterSI       ;(no) go to SetIterSI

SetIterNo       CLRF        0x1D            ;clear F1D
                GOTO        SetIterEnd      ;go to MACRO end

SetIterSI       CLRF        0x1D            ;clear F1D
                BSF         0x1D,0          ;set bit 0 of F1D  => SI
                GOTO        SetIterEnd      ;go to MACRO end

SetIterMI       CLRF        0x1D            ;clear F1D
                BSF         0x1D,1          ;set bit 1 of F1D  => MI
                GOTO        SetIterEnd      ;go to MACRO end

SetIterLI       CLRF        0x1D            ;clear F1D
                BSF         0x1D,2          ;set bit 2 of F1D  => LI
                GOTO        SetIterEnd      ;go to MACRO end

SetIterEnd      NOP                         ;filler
                ENDM

; MACRO TO GENERATE A TONE ---------------------------------------------------

Tone            MACRO                       ;start of tone MACRO

                LOCAL       ToneOutLoop     ;define local variables
                LOCAL       ToneInLoop      ;

                MOVF        0x1E,0          ;load W w/ tone duration file
                MOVWF       0x1C            ;into F1C...

                BCF         0x05,2          ;force OA high

ToneOutLoop     MOVWF       0x08            ;and F08
ToneInLoop      DECFSZ      0x08,1          ;decriment F08
                GOTO        ToneInLoop      ;if not zero go to InTimeLoop
                CLRWDT                      ;clear watch dog timer
                DECFSZ      0x1C,1          ;decriment F1C
                GOTO        ToneOutLoop     ;if not zero go to OutTimeLoop

                BSF         0x05,2          ;force OA low

                ENDM

; MACRO TO VARY TONE DURATION ------------------------------------------------
```

```
ToneDur      MACRO                      ;start of tone duration MACRO

             LOCAL       ToneDurEnd     ;define local variable

             BTFSS       0x07,7         ;is IB accessed ???
             GOTO        ToneDurEnd     ;(no) go to MACRO end

             MOVLW       0x1F           ;load W with "1F"
             ADDWF       0x1E,1         ;add W to tone duration file

ToneDurEnd   NOP                        ;filler
             ENDM

;************************ BEGINNING OF SOURCE CODE ***************************


;********** INITIALIZE MICROCONTROLLER

             SetTRIS                    ;define I/O port directions

             MOVLW       0x3F           ;load W with "3F"
             MOVWF       0x1E           ;trans W to tone duration file

             BSF         6,4            ;turn off all decimal points
             BSF         6,5            ;
             BSF         6,6            ;
             CLRF        0x1D           ;turn off tone generation
             BSF         0x05,2         ;

Begin        SetTRIS                    ;define I/O port directions

             ToneDur                    ;check for change in
                                        ;tone duration

;********** ACQUIRE AND PROCESS NEW COUNT DATA

Length       SET         0x03
             TimeDelay   Length         ;short time delay

             CountShift                 ;make file room for new RTCC

             CountReady                 ;initiate count window

             CountWindow                ;count number of RTCC pulses

             CountAve                   ;average count values

Length       SET         0x1F           ;set Length
             TimeDelay   Length         ;delay to discharge cap in 555

;********** UPDATE DISPLAY DIGITS
```

```
                HexToDec                        ;convert hex RTCC to bcd

                MOVF            0xA,0           ;load W with units value
                MOVWF           0x8             ;preload F08 with units value
                LoadDispDig                     ;put units value on disp bus
DigPos          SET             0x00            ;set DigPos for units digit
                LatchDispDig    DigPos          ;latch units digit data

                MOVF            0xB,0           ;load W with tens value
                MOVWF           0x8             ;preload F08 with tens value
                LoadDispDig                     ;put tens value on disp bus
DigPos          SET             0x01            ;set DigPos for tens digit
                LatchDispDig    DigPos          ;latch tens digit data

                MOVF            0xC,0           ;load W with hundreds value
                MOVWF           0x8             ;preload F08 with hunds value
                LoadDispDig                     ;put hunds value on disp bus
DigPos          SET             0x02            ;set DigPos for hunds digit
                LatchDispDig    DigPos          ;latch hundreds digit data

                MOVF            0x10,0          ;load W with N data value
                SUBWF           0x11,0          ;subtract N from N-1
                BTFSC           3,2             ;is result equal to "0" ???
                GOTO            SIZero          ;(yes) go to zero routine
                BTFSS           3,0             ;(no) is the result negative?
                GOTO            SINeg           ;(yes) got to negative routine

SIPos           MOVLW           0x05            ;(no) load W with lower square
                GOTO            SIMerge         ;go to output routine
SIZero          MOVLW           0x08            ;load W with double square
                GOTO            SIMerge         ;go to output routine
SINeg           MOVLW           0x02            ;load W with upper square data
SIMerge         MOVWF           0x08            ;trans lower square data to
                                   ;F08
                LoadDispDig                     ;put square data on diplay bus

DigPos          SET             0x03            ;set DigPos to SI digit
                LatchDispDig    DigPos          ;latch SI digit data

                MOVF            0x18,0          ;load W with new MI data
                SUBWF           0x19,0          ;subtract old from new MI data
                BTFSC           3,2             ;is result equal to "0" ???
                GOTO            MIZero          ;(yes) go to zero routine
                BTFSS           3,0             ;(no) is the result negative?
                GOTO            MINeg           ;(yes) got to negative routine
MIPos           MOVLW           0x05            ;(no) load W with lower square
                GOTO            MIMerge         ;go to output routine
MIZero          MOVLW           0x08            ;load W with double square
                GOTO            MIMerge         ;go to output routine
MINeg           MOVLW           0x02            ;load W with upper square data
```

```
MIMerge         MOVWF           0x08            ;trans lower square data to
                                  ;F08
                LoadDispDig                     ;put square data on diplay bus

DigPos          SET             0x04            ;set DipPos to SI digit
                LatchDispDig    DigPos          ;latch MI digit data

                MOVF            0x1A,0          ;load W with new LI data
                SUBWF           0x1B,0          ;subtract old from new LI data
                BTFSC           3,2             ;is result equal to "0" ???
                GOTO            LIZero          ;(yes) go to zero routine
                BTFSS           3,0             ;(no) is the result negative?
                GOTO            LINeg           ;(yes) got to negative routine
LIPos           MOVLW           0x05            ;(no) load W with lower square
                GOTO            LIMerge         ;go to output routine
LIZero          MOVLW           0x08            ;load W with double square
                GOTO            LIMerge         ;go to output routine
LINeg           MOVLW           0x02            ;load W with upper square data
LIMerge         MOVWF           0x08            ;trans lower square data to
                                  ;F08
                LoadDispDig                     ;put square data on diplay bus

DigPos          SET             0x05            ;set DigPos to LI digit
                LatchDispDig    DigPos          ;latch LI digit data

;********** UPDATA DECIMAL POINT DISPLAYS

                SetIterMode                     ;survey IA for change of
                                                ;iteration mode request

                BSF             0x06,4          ;clear all dec points
                BSF             0x06,5          ;
                BSF             0x06,6          ;

DigPos          SET             0x04            ;set dec pnt pos for SI
                BTFSC           0x1D,0          ;turn on SI dec point ???
                BCF             6,4

DigPos          SET             0x05            ;(no) set dec pnt pos for MI
                BTFSC           0x1D,1          ;turn on MI dec point ???
                BCF             6,5

DigPos          SET             0x06            ;(no) set dec pnt pos for LI
                BTFSC           0x1D,2          ;turn on LI dec point ???
                BCF             6,6

;********** GENERATE TONES AS REQUIRED

                BTFSC           0x1D,0          ;is SI tone enabled ???
                GOTO            SITone          ;(yes) jump to SITone
                BTFSC           0x1D,1          ;(no) is MI tone enabled ???
```

```
                GOTO        MITone       ;(yes) jump to MITone
                BTFSC       0x1D,2       ;(no) is LI tone enabled ???
                GOTO        LITone       ;(yes) jump to LITone
                GOTO        NoTone       ;(no) goto NoTone

SITone          MOVF        0x10,0       ;load W with N data
                SUBWF       0x11,0       ;subtract N from N-1 data
                BTFSC       0x03,2       ;is the zero bit set ???
                GOTO        NoTone       ;(yes) go to NoTone
                BTFSS       0x03,0       ;(no) is the carry set ???
                GOTO        NoTone       ;(no) go to NoTone
                GOTO        YesTone       ;(yes) go to YesTone

MITone          MOVF        0x18,0       ;load W with new MI data
                SUBWF       0x19,0       ;subtract new from old MI data
                BTFSC       0x03,2       ;is the zero bit set ???
                GOTO        NoTone       ;(yes) go to NoTone
                BTFSS       0x03,0       ;(no) is the carry set ???
                GOTO        NoTone       ;(no) go to NoTone
                GOTO        YesTone      ;(yes) go to YesTone

LITone          MOVF        0x1A,0       ;load W with new LI data
                SUBWF       0x1B,0       ;subtract new from old LI data
                BTFSC       0x03,2       ;is the zero bit set ???
                GOTO        NoTone       ;(yes) go to NoTone
                BTFSS       0x03,0       ;(no) is the carry set ???
                GOTO        NoTone       ;(no) go to NoTone
                GOTO        YesTone      ;(yes) go to YesTone

YesTone         Tone                     ;jump to Tone macro

NoTone          NOP                      ;filler

Length          SET         0x03         ;set variable value
                TimeDelay   Length       ;time delay

                GOTO        Begin

                END
```

Intentionally Left Blank

No. of Copies	Organization
2	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	Commander U.S. Army Materiel Command ATTN: AMCAM 5001 Eisenhower Ave. Alexandria, VA 22333-0001
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-SD-TA, Records Management 2800 Powder Mill Rd. Adelphi, MD 20783-1145
3	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-SD-TL, Technical Library 2800 Powder Mill Rd. Adelphi, MD 20783-1145
1	Director U.S. Army Research Laboratory ATTN: AMSRL-OP-SD-TP, Technical Publishing Branch 2800 Powder Mill Rd. Adelphi, MD 20783-1145
2	Commander U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000
1	Director Benet Weapons Laboratory U.S. Army Armament Research, Development, and Engineering Center ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	Director U.S. Army Advanced Systems Research and Analysis Office (ATCOM) ATTN: AMSAT-R-NR, M/S 219-1 Ames Research Center Moffett Field, CA 94035-1000

No. of Copies	Organization
1	Commander U.S. Army Missile Command ATTN: AMSMI-RD-CS-R (DOC) Redstone Arsenal, AL 35898-5010
1	Commander U.S. Army Tank-Automotive Command ATTN: AMSTA-JSK (Armor Eng. Br.) Warren, MI 48397-5000
1	Director U.S. Army TRADOC Analysis Command ATTN: ATRC-WSR White Sands Missile Range, NM 88002-5502
1	Commandant U.S. Army Infantry School ATTN: ATSH-WCB-O Fort Benning, GA 31905-5000
	Aberdeen Proving Ground
2	Dir, USAMSAA ATTN: AMXSY-D AMXSY-MP, H. Cohen
1	Cdr, USATECOM ATTN: AMSTE-TC
1	Dir, USAERDEC ATTN: SCBRD-RT
1	Cdr, USACBDCOM ATTN: AMSCB-CII
1	Dir, USARL ATTN: AMSRL-SL-I
5	Dir, USARL ATTN: AMSRL-OP-AP-L

No. of Copies	Organization
12	Dir, USARL ATTN: AMSRL-WT-WD, Dr. P. Berning Dr. R. Bossoli Mr. W. Clay Dr. S. Cornelison Dr. A. Gauss, Jr. Dr. C. Hollandsworth Dr. C. Hummer Dr. M. McNeir Dr. A. Niiler Dr. J. Powell Dr. A. Prakash Dr. G. Thomson

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number ARL-MR-180 Date of Report September 1994

2. Date Report Received ____________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ____________________

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ____________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ____________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ____________________

CURRENT ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 0001, APG, MD

Postage will be paid by addressee

Director
U.S. Army Research Laboratory
ATTN: AMSRL-OP-AP-L
Aberdeen Proving Ground, MD 21005-5066